BUILDING COLLABORATIVE KNOWLEDGE-BASED SYSTEMS: A WEB ENGINEERING APPROACH

George Gkotsis
CTI, E-Learning Sector
26500 Rio-Patras, Greece
gkotsis@cti.gr

Christina E. Evangelou
CTI, E-Learning Sector
26500 Rio-Patras, Greece
chriseva@cti.gr

Nikos Karacapilidis
Industrial Management and Information Systems Lab, MEAD
University of Patras, 26500 Rio-Patras, Greece
nikos@mech.upatras.gr

Manolis Tzagarakis
CTI, E-Learning Sector
26500 Rio-Patras, Greece
tzagara@cti.gr

ABSTRACT

Knowledge Based Systems (KBSs) has been a research and practice area of growing interest for more than a decade. After an initial success, the lack of sophisticated methodologies or theories for the transformation of domain knowledge in a form that is utilizable by a computer-based application, in addition to serious difficulties such as the low reusability of their components due to requirements imposed by diverse knowledge domains and groups, made the reconsideration of the KBSs development process appearing as a prominent need. On the other hand, Web Engineering is an emerging discipline that aims at the deployment of tools, techniques and methodologies to support the development of web applications. Taking the above remarks into account, this paper proposes a framework of practices for the development of collaborative KBSs that comply with it. The proposed framework deploys a series of tools and methodologies to support the development of the above applications during their whole life cycle, striving for openness, scalability and cross-platform compatibility. Much attention is also paid to issues related to requirements analysis and services specification with respect to a CoP's practices, the applications' architecture, and the related ontologies building.

KEYWORDS

Knowledge Management, Web Engineering, Communities of Practice.

1. INTRODUCTION

Experiences reported by an increasing number of companies show that their long-term survival and competitive success is determined not so much by their financial muscles and size, but by the manner in which they consciously attempt to learn, create, codify, and utilise knowledge. Consequently, a variety of computer-based frameworks, widely known as Knowledge-Based Systems (KBSs), have emerged in order to deal with the acquisition, storage and intelligent processing of the employees' knowledge, as well as with the process of making this knowledge accessible to their peers for the facilitation of diverse organizational activities [1]. According to the related literature, the most efficient involvement of humans in knowledge sharing activities enabled by on-line interactive technologies is through the establishment of virtual Communities of Practice

(CoPs) [2]. At the same time, given the central role of collaboration in knowledge sharing, the development of web-based collaboration tools has become an especially interesting topic for knowledge-based applications since these can facilitate and empower even international team members.

The recent trend towards more collaboration-oriented KBSs that integrate a series of (usually discrete) functions challenges for more sophisticated technical solutions. It is widely admitted that web-based tools can facilitate knowledge workers in accessing, analyzing, and further elaborating information. Web-based technologies exhibit a major impact on the design and implementation processes of all types of Information Systems. Compared to traditional batch-based or client-server oriented tools, they exhibit ease-of-use, universal access across information technology platforms, as well as single minute response and feedback based upon dynamic and real-time data. Furthermore, recent developments in the area of Web Engineering make the management of a complex system's interactions more and more efficient [3].

Remarks drawn from our experience in developing knowledge-based collaboration tools for diverse CoPs (e.g. [4]) identify the following issues:

- Knowledge *per se* is intensively domain dependent whereas KBSs are context specific applications. Thus, reusability is certainly a ubiquitous and complex issue.
- The lack of sophisticated methodologies or theories for the extraction of reusable knowledge and reusable knowledge management patterns has proven to be extremely costly, time consuming and error prone;
- Supporting knowledge sharing processes in a virtual community needs more technical support to deal with the asynchronous addition of knowledge objects and the maintenance of a consistent knowledge structure.
- Taking the above remarks into account, this paper proposes a framework of practices for the development of collaborative KBSs that comply with the Web Engineering discipline. The proposed framework deploys a series of tools and methodologies to support the development of the above applications during their whole life cycle, striving for openness, scalability and cross-platform compatibility.

On the other hand, Web Engineering has been defined as "the application of systematic, disciplined and quantifiable approaches to development, operation, and maintenance of Web-based Information Systems. It is both a pro-active approach and a growing collection of theoretical and empirical research in Web application development" [5]. The continuous grow of the web renders it not only as the most prominent way of publishing information, but also as an uprising and effective platform for supporting all facets of organizational work.Web engineering consists of an attempt to provide theoretical but basically practical armoury to both knowledge workers and developers. This is to be taken into account together with the vision for the Semantic Web, which is "an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation" [6]. From an engineering perspective, web engineering has been supported with several technologies. One of its main aspects is the notion of *web services*. Web services have been specified with standards, making it feasible to create heterogeneous applications which call upon different services provided by different manufacturers.

2. THE PROPOSED FRAMEWORK

In our perspective, developing KBSs that facilitate collaboration is about supporting and enhancing the organizational processes of knowledge creation, storage, retrieval, transfer, processing and application. Towards this aim, we develop a set of services and associated scenarios of use, whose design and implementation are based on real life CoPs' activities. Our overall development process is iterative, comprising five core tasks (see Fig. 1). In the beginning, we identify the services required from a particular CoP in order to address its specific needs as far as collaboration and communication are considered. After the formal definition of a CoP's requirements in terms of services, we formalize the CoP's practices, as well as their knowledge domain using a set of ontology models. We then implement the service-oriented components of the system. Finally, we proceed to the synthesis and specialization of the integrated services according to the particular CoP. In the following, we present the abovementioned processes in more detail, putting emphasis on the technical solutions to be employed.

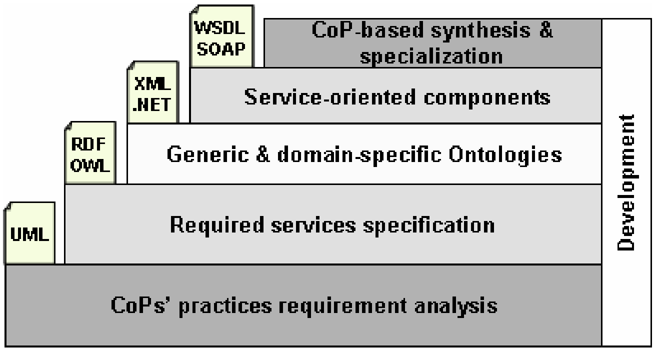

Figure 1. The proposed development process

2.1 CoPs practices requirements analysis

Generally speaking, the majority of organizational functions require the collaboration of a group of stakeholders (i.e. decision makers, domain experts, knowledge workers), often representing diverse practices and domains. Organizational forms created for these purposes are cross-functional teams and workgroups, recently known as Communities of Practice [7]. In most cases, the practices of such groups are facilitated by e-mail exchange, electronic discussion groups, and electronic chat rooms. Nevertheless, the need for developing methods and tools that support data production, exchange and reuse has increased over the years, especially with the proliferation of Web data sources deploying a variety of modalities, information models and data encoding syntaxes. Thus, our approach aims at developing interoperable tools that facilitate the fundamental activities of a CoP, respecting its particularities in terms of group membership and domain of practice.

Consider, for instance, that one wants to address the needs of three distinct types of CoPs (let them be managers, engineers and medical doctors) that follow completely diverse practices. These require a series of (usually discrete) services such as decision making support, argumentation facilitation, and negotiation. According to our approach, the requirements analysis of a multipurpose platform that provides integrated services is based on *participative design*, in order to fully support the CoPs real-life practices. More specifically, we identify the practices and the users' characteristics by observing and formalizing the CoP's individual and social activities. We then shape the services to be provided in the specific context (i.e. specific CoP, specific knowledge-domain).

2.2 Proposed Architecture

Most common practices from the Web Engineering discipline have been closely related to component-based software engineering. Software components are "binary units of independent production, acquisition, and deployment that interact to form a functioning system" [8]. Components can be called upon to allow the creation of arbitrary abstraction layers. Thus, the application needs can be adapted in a feasible way to the system supported by software components. According to our approach, components interact to each other by making use of web services. Web services can be described by making use of the well-known standard of Web Service Description Language (WSDL).

The process of building an application that conforms to a service-oriented architecture differs from traditional application development, since the latter needs to be integrated in a distributed environment and cooperate with heterogeneous software artifacts. In order to tackle the above issue, web applications need to define their functional and non-functional capabilities in a machine-readable format.

In our approach, applications constitute clients that share (to some degree) the same functionality. In order to eliminate time and cost effort, a three-tier architecture is adopted. On the bottom layer, the core services are designed to provide the common functionalities required by collaborative KBSs.

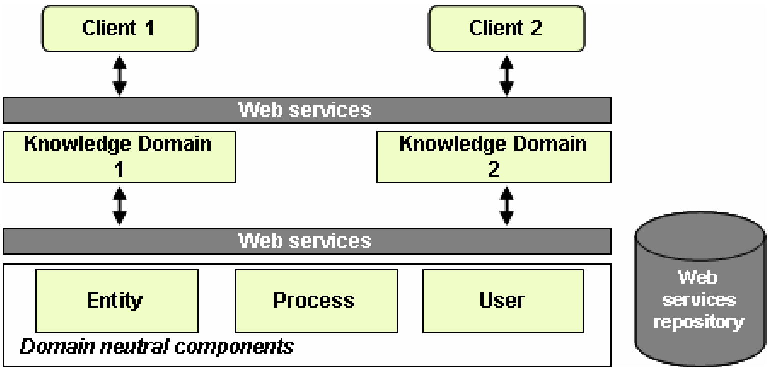

Figure 2 The proposed architecture

As shown in Fig. 2, the proposed overall architecture of a knowledge-based collaborative system comprises three layers. In the bottom layer, the *entity, process* and *user* components comprise the domain-independent components. We acknowledge that defining the set of the basic services provided by these components consists an ongoing process. Nevertheless, in the *entity* component we prescribe that the basic managing mechanisms will deal with storage, access and modifications of abstract entities. These entities are designed to act as containers for any domain-oriented entity (e.g. the entity `cure` in a medical CoP will be a specialization of the `solution` abstract entity class). The *process* component is intended to address the various activities taken place in any collaboration context (e.g. the process `supportCure` could be a specialization of a `supportArgument` abstract process). The *user* component models the actions taken place by the various persons participating in a collaborative environment. In this way, all the above components provide diverse services that are supplementary to each other concerning the domain-specific problem needs. Finally, we prescribe the *web services repository* component, which acts as the services registry of the system in order to facilitate the automated service discovery mechanism.

In order to better support the design and specification of the above services, we propose the design of the required services can be accompanied with well-known CASE tools. For instance, UML class diagrams can be used to identify the basic entities at their corresponding component, whereas use case, sequence, state transition and collaboration UML diagrams can be used to outline the various user types and activities for the remaining components.

2.3 Generic vs. domain-specific ontology building

Ontologies are a powerful means to accomplish a shared understanding of different knowledge domains and to facilitate sharing and reuse of bodies of knowledge across groups and applications [9]. Moreover, they figure prominently in the emerging Semantic Web as a way of representing the semantics of documents and enabling these semantics to be used by web applications and intelligent agents. Thus, in our approach, we employ ontology models that comprise a *generic* as well as a *domain-specific* part in order to properly handle concepts and models from various knowledge domains. The former regards the notions and practices of a CoP independently of the knowledge or problem domain considered. The latter is used to describe issues related to the particular context of use. The ontology models developed are formally defined with the use of an XML Schema model for interoperable, generic, extensible and neutral information modelling, which provides the semantic means to specify diverse knowledge domains and achieve a common understanding. Furthermore, they comply with the OWL language semantics, establishing a high-level of machine interpretability of Web content.

For instance, decision making issues can be formally defined by a generic part that describes besides the decision *per se*, the decision makers and the decision making process. As far as the decision makers are concerned, we formalize their personal and professional information, as well as notions related to their behaviour when participating in the related activities (e.g. frequency of participation, preferences of use). The generic part of a decision making process refers to the knowledge representation and processing practices employed by a CoP in order to a problem, attain a goal or seize an opportunity.

2.4 Service composition and specialization

In our framework, the adaptation of the system to a variety of CoPs is based on the ability to compose and specialize services. In order to achieve the above goal, we adopt the work described in [10], adapting it to the architecture described in Section 3.2. The platform-independent web services of the bottom layer of the proposed architecture can be specialized and composed in order to supply a set of web services required by a particular CoP.

The middleware of our architecture, which consists a domain-dependent component, can be described as a *service component* class (we use here the term “class” in order to stress the fact that the object-oriented paradigm is supported in our approach). More specifically, the service composition feature can be resolved by making use of two different approaches. The first approach is to define a new web service and call the required services in the implementation level. The second approach is to extend the abstract services in order to adapt them to the domain-specific context, following an XML specification for the service component, as described in [10].

3. CONCLUSIONS

We have argued that the proper development of collaborative knowledge-based systems does not only rely in identifying the practices of the organization to be employed, but it should also exhibit the virtues of extensibility, reusability, tailorability and neutrality. Towards this aim, we presented a framework that has been derived from our experience in developing software for the managing of the organizational knowledge embedded in collaborative settings. The proposed framework is in accordance with the Web Engineering discipline. We envisage it not just as another software design methodology, but as a practical guide that provides a structured way for developing complex KBSs. Our future work directions concern the thorough evaluation of our approach in diverse organizational settings and knowledge domains.

ACKNOWLEDGMENTS

Research carried out in the context of this paper has been partially funded by the EU PALETTE (Pedagogically Sustained Adaptive Learning through the Exploitation of Tacit and Explicit Knowledge) Integrated Project (IST FP6-2004, Contract Number 028038).

REFERENCES

[1] Meso, P. and Smith, R.: A Resource-based View of Organizational Knowledge Management Systems. Journal of Knowledge Management Vol. 4(3), 224–234 (2000).
[2] Liedtka, J.: Linking competitive advantage with communities of practice. Journal of Management Inquiry Vol. 8(1), 5–16 (1999).
[3] Murugesan S. and Deshpande, Y.: Web Engineering : Managing Diversity and Complexity of Web Application Development (2001).
[4] Evangelou, C.E., Karacapilidis, N., Abou Khaled, O. and Chabbi Drissi, H.: On the Elicitation of Knowledge in Collaborative Decision Making Settings. In Proceedings of: The ECKM 2005 Conference, Limerick, Ireland, 8-9 September (2005).
[5] Deshpande, Y., Murugesan, S., Ginige, A., Hansen, S., Schwbe, D., Gaedke, M. and White, B. Web Engineering. Rinton Press Journal of Web Engineering Vol. 1(1), 3-17 (2002).
[6] Berners-Lee, T., Hendler, J. and Lassila, O.: The Semantic Web, Scientific American, May (2001).
[7] Wenger, E., McDermott, R. and Snyder, W.M.: Cultivating Communities of Practice. Harvard Business School Press, Boston, MA (2002).
[8] Szyperski, C.: Component Software, Addison-Wesley, (1998).
[9] Chandrasekaran, B., Josepheson, J. and Benjamins, V.R.: Ontologies: What Are They? Why Do We Need Them? IEEE Intelligent Systems Vol. 141, 20-26 (1999).
[10] Jian Yang , Mike P. Papazoglou, Web Component: A Substrate for Web Service Reuse and Composition, Proceedings of the 14th International Conference on Advanced Information Systems Engineering, p.21-36, May 27-31, (2002).